United States
Environmental Protection Agency

National Environmental Scientific Computing Center
National Technology Services Division
Office of Technology Operations and Planning
Office of Environmental Information
Research Triangle Park, NC 27711

September 29, 2004

EPA

National Environmental Scientific Computing Center User Guide

Version 9.2

Table of Contents

1 Quick Reference

1.1 Help resources

EPA Call Center: Monday-Friday, 6:00 am -- 6:30 pm (ET): 866-411-4372
NESC^2 E-mail help: help@nesc.epa.gov
NESC^2 web page: http://www.epa.gov/nesc
Comments on this guide: anderson.edward@epa.gov

1.2 HPC systems

IBM eServer Cluster 1600 (2 login nodes):

```
Name: emerald00.nesc.epa.gov;  IP address: 134.67.68.12
Name: emerald01.nesc.epa.gov;  IP address: 134.67.68.13
```

1.3 EPA service managers

John B. Smith
Service Manager for High Performance Computing and Scientific Visualization
Infrastructure Support
National Technology Services Division
Office of Environmental Information
U.S. Environmental Protection Agency
Mail code: N229-01
Research Triangle Park, NC 27711
E-mail: smith.johnb@epa.gov
Telephone: 919-541-1087

Heidi K. Paulsen
Service Manager for High Performance Computing and Scientific Visualization
Project Support
National Technology Services Division
Office of Environmental Information
U.S. Environmental Protection Agency
Mail code: N229-01
Research Triangle Park, NC 27711
E-mail: paulsen.heidi@epa.gov
Telephone: 919-541-1834

1.4 Location of NESC^2/SVC

National Computer Center
U.S. Environmental Protection Agency
109 T.W. Alexander Drive
Research Triangle Park, NC 27711

Shipping address:

U.S. Environmental Protection Agency
Attn: (staff name here) (N127-01)
4930 Page Road
Durham, NC 27703

2 What is NESC^2/SVC?

The National Environmental Scientific Computing Center (NESC^2) is an EPA managed, contractor operated facility designed to support EPA's high-performance computing (HPC) programs. It includes the Scientific Visualization Center (SVC), a group of dedicated visualization specialists who work with scientists and analysts to gain greater insights into the results of their research and to effectively convey those results to others. NESC^2 is located at EPA's National Computer Center (NCC) in Research Triangle Park (RTP), North Carolina.

2.1 Services of the NESC^2

NESC^2 provides Agency customers with high performance computing, scientific information management, and scientific visualization services, resources, and expertise to support their progress toward Agency goals.

In support of this mission, NESC^2 staff provide infrastructure support and direct customer support to help scientists develop models that effectively and efficiently use the HPC resources to solve their most complex computational problems. NESC^2 staff maintain the HPC systems in accordance with industry best practices, customize system interfaces to meet the needs of NESC^2 projects, install software, and manage the data archive. In a computational science consulting role, NESC^2 staff assist customers with troubleshooting, code porting and optimization, provide expertise in scalable algorithm design, and conduct user training. NESC^2 staff also monitor system performance, assess capacity needs, and evaluate new technologies to ensure that NESC^2 provides a reliable, responsive, scalable, flexible, and cost-effective computing solution for its customers.

2.2 Services of the SVC

The mission of the SVC is to assist researchers in extracting the entire value of numerical data, whether it be empirical data gathered from observers or sensors or derived data generated by computational models, through the visualization of that data.

In support of this mission, SVC staff provide state-of-the-art visualization resources that assist and advance the research work of EPA regions, program offices, and laboratories. Through collaboration with researchers, SVC staff explore, evaluate, and test new methods for scientific and information visualization and develop the best of these methods into working prototypes. Furthermore, SVC staff transfer this new knowledge to its customers by providing training, developing and presenting special demonstration projects, and by providing expertise and guidance in the specification, procurement and implementation of user-owned visualization systems and/or labs.

2.3 Resources

At the start of FY2005, NESC^2 supported the following resources:

I. Computational Resources

- IBM eServer Cluster 1600 (128 processors, 11.5 TB disk)

II. Data Management Resources

- Sun E4500 File Server
- STK Tape Silos
- >200 TB Archival Tape Storage

III. Visualization Resources

- SGI Onyx3400 (8 processors, 218 GB disk, InfiniteReality3 graphics)
- SGI Onyx2 (8 processors, 144 GB disk)
- Media-100 digital editing system
- Large-format, high-definition display system

2.4 Getting a NESC^2 account

Accounts on the NESC^2 HPC systems are primarily available to EPA researchers, EPA contractors, and individuals affiliated with EPA-sponsored projects. Trial accounts for the purpose of evaluating the performance and capability of the NESC^2 HPC systems may also be granted on a limited basis at the direction of the Service Manager (see Section 1.3 on page 5).

First-time users of the EPA network will need to complete a network access agreement and obtain a 3-character User ID from their ADP coordinator. If you don't know who your ADP Coordinator is, call the EPA Call Center at 1-866-411-4EPA (4372) for assistance. Further information is available at the following EPA web page:

`http://www.epa.gov/webguide/started/getserv.html`

Applications for NESC^2 projects are solicited in early spring, subjected to peer review, and evaluated by the High Performance Computing Working Group (HPCWG). Allocations of system resources to project accounts typically begin at the start of the fiscal year (October 1). New users on an existing project can be added at any time at the direction of the Service Manager. Application forms and instructions are available at the following EPA NESC^2 web page:

`http://www.epa.gov/nesc/00_general/administration.html`

Trial accounts for evaluation purposes may be requested from the Service Manager. To apply for a trial account, please send the following data about yourself and project to the Service Manager via EPA, Internet, or U.S. mail:

1. Name
2. EPA 3-character login ID
3. U. S. Mail address
4. EPA E-mail address (if available)
5. Internet E-mail address (if available)

6. Phone number
7. A brief description, in layman's language, of the application for which you need the supercomputer. Please limit this description to a one page narrative without mathematical formulas.

2.5 Accessing NESC^2 HPC systems

Users inside the EPA firewall should be able to access the NESC^2 systems directly using ssh or telnet (ssh is preferred) to one of the following addresses. The addresses of the data servers are included even though most users do not have a need to access them directly.

IBM eServer Cluster 1600 (2 login nodes):

```
Name: emerald00.nesc.epa.gov;  IP address: 134.67.68.12
Name: emerald01.nesc.epa.gov;  IP address: 134.67.68.13
```

Sun E4500:

```
Name:  sweetgum.nesc.epa.gov; IP address: 134.67.64.21
Name:  walnut.nesc.epa.gov;  IP address: 134.67.64.17
```

Users outside the EPA firewall must first be authenticated for access through the firewall. The most commonly used method is the Anytime Anywhere Access Service (http://intranet.epa.gov/aaa), which uses the Aventail Connect Client from a Windows PC. The steps of this process are as follows:

1. Fill out an application for Anytime Anywhere Access, available at

   http://intranet.epa.gov/aaa/aaa_howtoorder.htm

   Once approved, you will be sent a SecurID card which generates random numbers used to construct a unique single-use password each time you log in.

2. Get your username/SecurID authenticated for the firewall, with assistance from the EPA Call Center (1-866-411-4372).

3. Log in at https://shield.epa.gov to access the EPA Intranet.

4. Download and install the Aventail connect client from

   https://shield.epa.gov:443/intranet/aaa/aaa_anywherevpn.htm

5. If necessary, install an ssh client on your PC. This can be a client of your choice; one option is an application called PuTTY, available for download at

   http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

6. Check your personal firewall settings. You shouldn't have to disable your firewall, but you may need to adjust its settings to a moderate level of security to allow communication with the EPA firewall.

7. Start up the Aventail client.

8. With the Aventail client running, open an ssh session to one of the NESC^2 systems, for example, `emerald00.nesc.epa.gov`. The Aventail client will intercept traffic from the ssh port and direct it through the EPA firewall.

If you have any questions regarding Aventail access, contact the EPA call center at 1-866-411-4372.

2.6 Putting your data on the system

Most users transfer their program source code and much of their data electronically using ftp, sftp, or scp. Further details about scp are in Section 3.5 on page 12. Typical transfer rates across the internal EPA gigabit Ethernet network are about 1 GB per minute. External users can also use the EPA's authenticated science ftp server (`http://scienceftp.epa.gov`). Transfer rates from outside the EPA domain vary widely, but tests with NOAA's National Center for Environmental Prediction have sustained about 8 GB per hour.

Alternatively, data may be sent to the NESC^2 on magnetic tape. NESC^2's archival storage system consists of StorageTek 4400 silos containing STK 9840 and STK 9940 tapes. Internally, NESC^2 can provide limited support for 1/4 inch cartridge tapes, 4mm DAT tape, Exabyte 8mm helical scan tapes, and DDS-3 tapes. Contact the NESC^2 staff for the latest availability status of these tape formats.

3 Unix Basics

All the NESC^2 systems run some "flavor" of the Unix operating system. The system-specific aspects of the user and programming environment are discussed in sections 4 through 5. This section discusses some basic functions that are common to all the NESC^2 systems.

3.1 Changing your password

When your account is first created, you will be given a user ID and single-use password. The first time you log in, you will be prompted to change your password. The password you choose must be at least 8 characters long, and it must contain at least one non-alphabetic character, such as a number. Passwords are case-sensitive. On emerald, a password change will be immediate on the particular node where you changed your password, but it may take up to an hour to propagate to the other nodes of the system.

There are three ways to change your password:

1. On your first login, and every 90 days thereafter, you will be prompted to change your password at login. After changing your password, you will be logged out and you'll have to log back in using your new password.

2. While logged in to one of the NESC^2 systems, you can enter the command `passwd`, and follow the prompts.

3. As a last resort, one of the NESC^2 system administrators can reset your password for you. For security reasons, the NESC^2 staff can not change your password over the phone. The procedure to request a password reset is

   a. Request a password reset from your High Performance Computing Working Group (HPCWG) representative;

   b. Your HPCWG representative will convey your request to the EPA Service Manager (see Section 1.3 on page 5);

   c. The EPA Service Manager will issue technical direction to the NESC^2 staff to reset your password;

   d. NESC^2 staff will reset your password and give the new password to the EPA Service Manager;

   e. The EPA Service Manager will give the new password to your HPCWG representative;

   f. Your HPCWG representative will give the new password to you.

If you remember your password, but have mis-typed it too many times, you may be locked out of your account. In this case, you don't need a new password. Contact the NESC^2 staff, and they will reset your invalid login count.

3.2 Setting your file permissions

On Unix systems, each user is responsible for his or her own file security by appropriately setting the file access mode of his or her files and directories. Read, write, and execute permissions can be set separately for the user, group, and others outside the user's group. The current file access mode can be viewed using the long form of the ls command. For example,

```
emerald00% ls -l /etc/environment
-rw-r--r--   1 root     system    2400 Mar 08 10:55 /etc/environment
```

The first character displays "`-`" for a file, "d" for a directory, and "`l`" for a link. The next 9 characters display the permissions for the user, group, and others, respectively, where "`-`" indicates that permission is not allowed and the letters "`r`", "`w`", and "`x`" indicate that read, write, and execute permissions, respectively, are granted. In the above example, the `/etc/environment` file is owned by the `root` user in the `system` group, and read permission is granted for the user, group, and others, but write permission is granted only for the root user.

Read permission for a file can be set to allow sharing of a file with your group or with anyone. Execute permission must also be set to allow access to a directory or executable program. Generally, write permission is enabled for the user only, but you can turn off user write permission to prevent yourself from inadvertently overwriting the file. Enabling write permission for anyone except yourself (the user) is not recommended because it would allow others to modify your files or create new files in your directory that you could not delete.

The permissions of a file can be changed by its owner using the `chmod` command. File permissions can be specified symbolically or via a 3-digit octal string. The following examples illustrate some common operations using `chmod` (see the `chmod` man page for further details):

`chmod g+r foo`
: Add read permission for the group to the file `foo`.

`chmod a+r *`
: Add read permissions for all (user, group, and others) to every file in the current directory.

`chmod og-rwx dir`
: Remove read, write, and execute permissions for the group and others from the file or directory `dir`.

`chmod 755 dir`
: Set read, write, and execute permissions for the user and read+execute permissions only for the group and others for the file or directory `dir`.

`chmod -R og+rX dir`
: Recursively modify permissions for the group and others to add read permission for all files and directories and execute permission for directories and executable files in `dir` and its contents.

3.3 Forwarding email

All the NESC^2 systems have the capability to send and receive electronic mail. This capability enables some useful notification features; for example, the batch job scheduler can send you mail when your job completes. However, you probably have a better mail program on the PC or workstation on your desk, and you should forward your mail there instead of using the Unix mail program to read it on the NESC^2 system. To have your mail forwarded, create a file called `.forward` in your home directory containing your email address, for example:

```
leavitt.michael@epa.gov
```

3.4 Secure shell (ssh)

Users are advised to use ssh to connect to the NESC^2 systems. SSH (Secure Shell) is a program for logging into a remote machine and for executing commands on a remote machine. It provides secure encrypted communication between two hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel.

The procedure for enabling X11 forwarding depends on the SSH client you are using. Here are the instructions for two common clients.

Unix system with OpenSSH. For X11 forwarding, add the -X option:

```
ssh -X emerald00.nesc.epa.gov
```

On emerald, you should not set the DISPLAY environment variable in your `.cshrc` or `.profile` file when connecting with `ssh -X`; the ssh client will set it

for you. If your login name on emerald is different than on the system you are connecting from, you will also need to specify the option "`-l uid`", or use

```
ssh -X uid@emerald00.nesc.epa.gov
```

where uid is your user ID. If you don't need X11 forwarding, the -X option may be omitted.

Windows system with PuTTY. Open the putty application by double-clicking it. Enter the name of the NESC^2 system (such as emerald00.nesc.epa.gov) in the Host Name field and select SSH (using port 22). In the Category box, go to Connection -> SSH -> Tunnels and click the box for "Enable X11 forwarding". Now go back to Session, give the session a name and save it if you like, and click Open to connect.

The following paragraph was copied from another site -- please send any comments or corrections to the NESC^2 staff. If your X windows emulator is Hummingbird Exceed, open the Xconfig from your Exceed folder. In "Screen Definition", change to "Multiple" Window mode and save it. Open your "Communication" icon and set the Startup mode to "Passive".

Once you've connected and started Exceed, you can test that the forwarding is working by running a simple X application on emerald such as xcalc or xclock. It should display on your screen.

3.5 Secure copy (scp)

Users are advised to use the secure copy command, `scp`, to transfer files between systems, instead of `ftp`. The `scp` command works just like the Unix `cp` command for copying one file to another except that one or both file names may be preceded by a system name followed by a colon, or by the designation `user@system:`. As with the standard Unix `cp` command, you can preserve the creation date of your file by adding the `-p` option, and you can copy a directory by adding the option `-r` for "recursive". Here are some examples:

`scp file1 file2`
: Copy `file1` to `file2` locally (same as `cp`)

`scp file1 emerald00:`
: Copy local file `file1` to your home directory on emerald00

`scp emerald00:/work/gah/file1 file2`
: Copy `/work/gah/file1` from emerald00 to the local file `file2`

`scp -p -r results jmuir@home:emerald_files`
: Copy the directory `results` and its contents, preserving file modification times, to the directory `emerald_files` on the system `home`, using the login name `jmuir`.

4 IBM eServer Cluster 1600 (emerald)

The main HPC system at NESC^2 is an IBM eServer Cluster 1600 with 16 pSeries 655 nodes, each with eight 1.5 GHz IBM POWER4+ processors and 16 GB of memory, connected by a high performance switch. The system, named “emerald” after the North Carolina state gem, has a combined peak capacity of 768 Gflop/s and contains 16 terabytes (TB) of disk space using IBM TotalStorage FastT disk technology. Emerald was installed in February, 2004.

The IBM eServer Cluster 1600 has 16 nodes named emerald00 through emerald15. Two nodes, emerald00 and emerald01, are dedicated to interactive use; these are the only nodes you can log in to from the EPA network. The remaining nodes, emerald02 to emerald15, are batch nodes used for running jobs through the batch queueing system, LoadLeveler (see Section 4.5 on page 20 for more details about LoadLeveler).

User file space on emerald consists of a large (16 TB) shared file system called `/gpfs` which is managed by a file system quota implemented by groups (see Section 4.3 on page 15). All the nodes can access files in `/gpfs`, but only the two interactive nodes can access the network-attached `/asm` file system (see Section 5 on page 25 for more details about `/asm`). Consequently, any files from `/asm` that are needed on one of the batch nodes must first be copied to the local `/gpfs` disk, and any output files must be created on `/gpfs` and copied to `/asm` at the conclusion of the job.

4.1 Operating system

The operating system on the IBM eServer is AIX, currently version 5.2. The command to check the versions of installed system software is `lslpp`; for example, to see the version of the base operating system, the command is:

```
emerald01% lslpp -L | grep "bos.mp "
  bos.mp             5.2.0.19    C     F    Base Operating System
```

4.2 User environment

Several different Unix shells are available for use on the IBM eServer. New accounts are created using the C shell by default, but you can change your shell using the command `chsh` if desired. The command will show you a list of available shells. It will take up to 20 minutes for a change to propagate to the other nodes of the system.

Default settings for the PATH and MANPATH environment variables for all users are found in the system file `/etc/environment`. If you want to add to these paths, be sure not to overwrite the system settings. For example, to add the netCDF I/O package to your search path every time you log in, you could put the following lines in your `.cshrc` file (for C shell users):

```
setenv NETCDF_PATH /usr/local/netcdf-3.5.1
setenv PATH "${PATH}:${NETCDF_PATH}/bin"
setenv MANPATH "${MANPATH}:${NETCDF_PATH}/man"
```

or the following lines in your .profile file (for K shell users):

```
NETCDF_PATH=/usr/local/netcdf-3.5.1
export PATH="${PATH}:${NETCDF_PATH}/bin"
export MANPATH="${MANPATH}:${NETCDF_PATH}/man"
```

Some third-party software packages have been installed under Modules to simplify the setting of environment variables. The Modules package is a set of scripts and information files that provides a simple command interface for modifying the user environment. Each module in the Modules package is a file containing the information needed to initialize the environment for a particular application. The commands to load a module are independent of the shell that the user may be running. Modules are particularly useful for maintaining more than one version of an application, and the *module* command greatly simplifies the process of switching from one version of an application to another.

In order to access the features of Modules, you must first run an initialization script. This is typically done by sourcing a script in your `.cshrc` file (for C shell users) or `.profile` file (for K shell users). For C shell users, the lines to put in your `.cshrc` file are

```
if (-f /etc/modules/csh ) then
   source /etc/modules/csh
   module load modules
endif
```

while for K shell users the lines to put in your `.profile` file are

```
if [ -f /etc/modules/ksh ]; then
   . /etc/modules/ksh
   module load modules
fi
```

Some of the commonly used module commands are

`module avail`
: List all available modulefiles in the current MODULEPATH

`module list`
: List all loaded modules

`module display <modulefile>`
: Display changes modulefile will make to the environment

`module load <modulefile>`
: Load modulefile into the shell environment

`module switch <oldmodulefile> <newmodulefile>`
: Switch loaded oldmodulefile with newmodulefile

`module rm <modulefile>`
: Remove modulefile from the shell environment

`module help <modulefile>`
: Print any module specific help information for the modulefile

For example, the command to add netCDF to your search path using Modules is as follows:

```
module load netcdf
```

4.3 User file systems

On emerald, most of the user file systems are part of IBM's General Parallel File System (GPFS) and are shared by all the nodes. The GPFS file systems on emerald are

`/gpfs/home`
: GPFS file system for user home directories.

`/work`
: GPFS file system for user work directories.

`/ptmp`
: GPFS file system for temporary files. All files in /ptmp are automatically deleted after 30 days.

There is also a very small non-shared file system (only 256 MB) called `/tmp` on each node, but this space is needed by the system and should not be used as a work area. The archival file system, `/asm`, is accessible only from emerald00 and emerald01. The `/asm` file system is discussed in more detail in Section 5 on page 25.

The GPFS file system is managed by quotas specified by group, where a group corresponds to a NESC[2] project. Allocations of disk space for each project are made yearly based on the storage requirements specified in the project's resource request form and availability. Each project has a corresponding Unix group, usually of the same name; for example, the Unix group for the LUNAR project is named "lunar". All the users assigned to a project share its allocation. An individual user may be assigned to more than one project, in which case he or she shares a disk space allocation with more than one group. Each user who does work for more than one project is responsible for setting the group ownership of his or her files to match the appropriate project. The command to set the group ownership of a file is `chgrp`; for example, the command to change the group ownership of a directory called "lunarfiles" and all its contents to "lunar" is

```
chgrp -R lunar lunarfiles
```

The disk space allocation consists of a "soft" limit or quota, which can be exceeded for up to 30 days, and a "hard" limit, which can not be exceeded. The hard limit is generally about 110% of the quota. If a group exceeds its disk space quota, every member of that group will receive an automatically generated reminder message asking him/her to clean up his/her files, including those users whose total usage is zero. The purpose of these weekly reminders is to encourage members of the group to talk among themselves and to manage their allocation equitably. There is no automatic migration of files; if a group exceeds its quota for more than 30 days, or the group's usage reaches its hard limit, no one in the group will be able to create any more files. Files in `/gpfs/home`, `/work`, and `/ptmp` all count towards the quota.

Here are some useful commands for managing your quota:

`groups`
: Shows the Unix groups enabled for your ID.

`myaccount`
: Shows the projects your ID is registered under.

`quota`
: Shows your personal file space usage on GPFS and the usage for each of your groups. This command is adapted from the “mmlsquota” command in AIX to better fit a typical 80-column window.

The quota command displays information similar to the following:

```
User block usage and limits
Filesystem    user        KB      quota      limit   in_doubt  grace
/gpfs          jmu    7260000          0          0       5120   none

User file usage and limits
Filesystem    user      files      quota      limit   in_doubt  grace
/gpfs          jmu       5156          0          0         20   none

Group block usage and limits
Filesystem   group        KB      quota      limit   in_doubt  grace
/gpfs       trees   33800000   31500000   35000000      14128 29days

Group file usage and limits
Filesystem   group      files      quota      limit   in_doubt  grace
/gpfs       trees      11430          0          0        72    none
```

Here user `jmu` from group `trees` is using 7260000 KB (or 7.26 GB) among 5156 files. His individual quotas show 0 for both disk space blocks and the number of files because quotas are only implemented by groups. His group is using 33.8 GB, which is over their quota of 31.5 GB but still less than their hard limit of 35 GB, and they have 29 days left of the grace period to bring their usage back under the quota. If this user were a member of more than one group, the “Group” output would show a one-line summary for each of his groups.

Other than deleting files you no longer need, the most useful means for managing your disk space quota is to move files to the archival file system, `/asm`.

4.4 Programming Environment

4.4.1 Fortran, C, and C++ compilers

The IBM Fortran, C, and C++ compilers are called `xlf`, `xlc`, and `xlC`, respectively, but there are many different ways to invoke them to include different combinations of compiler and loader options.

The Fortran commands are

`xlf`	Fortran 77 conformance (fixed form default)
`f77`	Alias for xlf
`fort77`	Another alias for xlf

`xlf90`	Fortran 90 conformance (free form default)
`xlf95`	Fortran 95 conformance
`xlf_r`	Fortran 77 with thread-safe components
`xlf90_r`	Fortran 90 with thread-safe components
`xlf95_r`	Fortran 95 with thread-safe components
`xlf_r7`	Thread-safe Fortran 77 using Draft 7 POSIX pthreads API
`xlf90_r7`	Thread-safe Fortran 90 using Draft 7 POSIX pthreads API
`xlf95_r7`	Thread-safe Fortran 95 using Draft 7 POSIX pthreads API
`mpxlf`	Fortran 77 with MPI
`mpxlf90`	Fortran 90 with MPI
`mpxlf95`	Fortran 95 with MPI
`mpxlf_r`	Thread-safe Fortran 77 with MPI
`mpxlf90_r`	Thread-safe Fortran 90 with MPI
`mpxlf95_r`	Thread-safe Fortran 95 with MPI

All of the Fortran commands implement the same language set, which supports the Fortran 95 standard, so the biggest difference is in the default source form. The `xlf` commands expect the 72-column format from Fortran 77, while the `xlf90` and `xlf95` commands expect code in free source format. The source format can also be selected by the compiler options `-qfixed` or `-qfree`. The default file extensions supported are `.f` for normal compilation and `.F` to automatically invoke the preprocessor. Additional file extensions are supported through command-line options; for example, files with the `.f90` extension can be compiled by specifying `-qsuffix=f=f90`.

Many of the compiler command variations are concerned with thread-safety. A thread is an execution unit with its own registers, stack, and process resources. A subroutine is termed “thread-safe” if it can be called safely from two or more concurrently running threads. Language features that save the current state of a program unit for reference in a subsequent call, such as SAVE statements and COMMON blocks in Fortran, are typically not thread-safe. Thread-safety is a requirement for shared memory programs and for MPI applications that use shared memory parallelism within a node.

There are also over 30 different ways to invoke the C/C++ compiler. Some of the most commonly used ones are

`xlc`	ANSI C compiler, UNIX header files
`cc`	C compiler, extended mode
`xlC`	C++ compiler
`xlc_r`	ANSI C compiler with AIX threads
`cc_r`	C compiler, extended mode, with AIX threads
`xlC_r`	C++ compiler with AIX threads
`mpcc`	C compiler with MPI
`mpcc_r`	C compiler with MPI and AIX threads
`mpCC`	C++ compiler with MPI
`mpCC_r`	C++ compiler with MPI and AIX threads

You can see which version of the compiler is installed by looking at the output of the `lslpp` command, for example:

```
emerald00% lslpp -L | grep "xlfrte "
  xlfrte          8.1.1.5    C    F   XL Fortran Runtime Environment
emerald00% lslpp -L | grep "vacpp\.cmp\.rte"
  vacpp.cmp.rte   6.0.0.0   C    F   VisualAge C++ Compiler
```

Fortran compiler configuration options are defined in the file `/etc/xlf.cfg` and C compiler configuration options are defined in `/etc/vac.cfg`.

4.4.2 Recommended compiler options

As you might expect, the compilers have many different options to assist with code development, debugging, and optimization. The following list briefly describes some of the options recommended by the NESC^2 staff:

`-O3 -qstrict`
: The `-O` options specify various levels of optimization; `-O3` by itself may alter the semantics of your program, such as changing (a*b)*c to a*(b*c), and the additional option `-qstrict` prevents this. The default is no optimization.

`-qmaxmem=8192`
: Provides more memory to the compiler if you're using `-O2` (not needed with `-O3`). The default `-qmaxmem` setting is 2048 KB at `-O2` and unlimited memory (`-qmaxmem=-1`) at `-O3`.

`-qfloat=nomaf`
: Suppresses the generation of multiply-add instructions. The multiply-add instructions are faster, but can cause some unexpected numerical results; for example, a*b - a*b may be relatively small but not zero.

If you're still debugging, the following additional options are useful (they will affect performance, so you probably should not use them in the production version of your code):

`-g or -qdbg`
: Produces information needed by the Totalview debugger.

`-C or -qcheck`
: Performs run-time checking of array bounds and character substring expressions (Fortran only).

`-qflttrap=overflow:zerodivide:invalid:enable -qsigtrap`
: Traps on floating-point exceptions, instead of continuing to compute with IEEE infinity and "not a number" (NaN) values. The first of these options may be abbreviated `-qflttrap=ov:zero:inv:en`.

You don't need to specify the following Fortran compiler options, which are enabled by default at NESC^2, but you may need to specify them for consistency on an identical platform elsewhere:

`-WF,-D_AIX`
: Set `_AIX` as a defined constant for the Fortran preprocessor (this constant is already set by default in the C environment).

`-qarch=auto`
Compile for the local machine.

`-qcache=auto`
Compile for the local cache size.

4.4.3 The 32-bit and 64-bit Application Programming Interfaces (APIs)

The default Application Programming Interface (API) in AIX is 32-bit, in which all pointers are 32-bits long and the addressable memory space is limited to 2 GB. This is sufficient for most MPI programs on emerald because the system has 16 GB of memory and 8 processors per node, which averages out to just under 2 GB per processor if all the memory is used. A 64-bit API is also supported via the compiler option `-q64`. This option changes the size of pointers to 64 bits and increases the limits for addressable memory, stack size, heap size, and file size for the application. The sizes of other variables, such as floating-point or integer variables, are not changed by the addition of the `-q64` flag.

When compiling and linking for the 64-bit API, the `-q64` option must be specified for every program unit of the application. If the link step includes any user-created libraries, they must be created with the `-X64` or `-X32_64` option to the `ar` command. Alternatively, the environment variable OBJECT_MODE may be set to 64 to automatically enable the appropriate options to compile and link for the 64-bit API.

4.4.4 The parallel operating environment (poe)

Execution of parallel programs is initiated by the IBM-specific `poe` command, which stands for *Parallel Operating Environment*. When running interactively, `poe` looks for a file containing a list of host names, one per process, specifying the node on which to run each process of the parallel job. For example, an interactive parallel invocation of the “hello, world” program might look like this:

```
emerald00% poe hello_mpi -hostfile hostfile -procs 8
```

where the file “hostfile” contains at least 8 lines such as

```
emerald00
emerald00
emerald00
emerald00
emerald00
emerald00
emerald00
emerald00
```

Alternatively, one can set the environment variable `MP_HOSTFILE` to the name of an existing host file and leave off the `-hostfile` option. The `poe` command and the hostfile are only needed when running interactively; in batch mode, the job script specifies how many parallel tasks should be used, and the batch job scheduler initiates the tasks wherever it can find free slots.

4.5 Batch queueing system - LoadLeveler

The batch queueing system on the IBM systems is called *LoadLeveler*. It is used for scheduling jobs and dividing the workload among all the nodes in the system.

4.5.1 LoadLeveler commands

The main LoadLeveler commands are

```
llsubmit <script>       : submit a job to the queues
llq                     : show the queued and running jobs
llstatus                : show status of the nodes
llclass                 : list the job classes (queues)
llcancel <requestid>    : cancel a job in the queues
```

The long forms of the `llq`, `llstatus`, and `llclass` commands using the `-l` option provide additional details about the jobs in the system and the configuration of the queues. A graphical interface to the LoadLeveler commands is also available using the X-interface command `xloadl`.

4.5.2 Job command file keywords

In order to submit a batch job to LoadLeveler, you need to create a job command file. The job command file contains LoadLeveler keyword statements that describe the resources required to run your job. One of these statements may specify the name of a binary file or shell script to be executed; otherwise the job command file itself is assumed to be a shell script and may include shell commands. The shortest possible job script would be something like the following:

```
#@ queue
env | grep "LOADL" > LL_env_vars
```

This example uses only the default settings in LoadLeveler and executes a serial job consisting of one shell command. However, if you want to run a parallel job, you need to specify some additional LoadLeveler keyword statements before the `#@ queue` line.

LoadLeveler keyword statements take the form

```
#@ keyword = value
```

A full description of the LoadLeveler keyword statements can be found in *IBM LoadLeveler for AIX 5L, Using and Administering, Version 3 Release 2*:

http://publib.boulder.ibm.com/clresctr/windows/public/llbooks.html

Some of the more commonly used keyword statements are as follows:

`#@ account_no = <account_name>`
Specifies the account to which this job's computing cycles should be attributed. Generally, account names are the same as project names. You can use the NESC^2-specific `myaccount` command to see which account names may appear on this line. If your userID is only active on one project, you may omit this line.

`#@ blocking = <integer>|unlimited`
Specifies how tasks of a parallel job are assigned to the nodes. If you don't care how the tasks are assigned, you can specify

```
total_tasks = <number>
blocking = unlimited
```

`#@ class = <class_name>`
Specifies the class (or queue) in which to run the job. Each project has its own class on emerald. There is also a special class called `single_proc` which can be used to run single-processor jobs on the interactive nodes (emerald00 and emerald01) to facilitate I/O processing. If your userID is only active on one project, you may omit this line.

`#@ cpu_limit = hh:mm:ss`
Sets a maximum CPU time limit for each process of the job step.

`#@ environment = var=value`
Specifies environmental variables to be set when your job step starts. You can set more than one environmental variable with this keyword by separating the assignments *var=value* with semicolons. If any of these variables are also set in your login initialization files , such as .profile or .cshrc, the settings in the initialization files will be used.

`#@ error = <filename>`
Specifies the name of a file to hold standard error output. By default, standard error is directed to /dev/null. A common technique is to save both standard error and standard output to file names that incorporate the jobid set by LoadLeveler, for example:

```
error = err.$(jobid)
output = out.$(jobid)
```

`#@ executable = <executable_file>`
For serial jobs, specifies the name of a program or script to run; for parallel jobs, `executable` is generally used only to specify a shell script. If this keyword is not specified, LoadLeveler assumes the job command file is the executable.

`#@ initialdir = <path>`
Specifies a starting directory for the job step. By default, the job begins execution in the directory from which it was submitted.

`#@ input = <filename>`
Specifies a file name for standard input, often used in conjunction with the `executable` keyword.

`#@ job_type = serial|parallel`
Indicates whether the job is serial or parallel. MPI jobs are always parallel, even if they use only one task. Serial is the default.

`#@ network.mpi = <type>[,<usage>[,<mode>[,<level>[,`
`instances=<number>|max]]]]`
Specifies the adaptors to use and characteristics for MPI communication. The *<type>* is required, but the other parameters are optional. The values for the parameters are as follows:

<type>
csss | sn_all: Use striped communication over all available switch networks

css0 | sn_single: Use a common, single switch network

<usage>
Specifies if the adapter can be shared with tasks of other job steps. Settings are either shared (the default) or not_shared.

<mode>
IP: Internet Protocol (default)

US: User Space, generally faster than IP, but may have to wait for an adapter window to become free

<level>
Specifies the amount of inter-task communication, either LOW, AVERAGE (the default), or HIGH.

instances=<*number*>|max
Number of parallel communication paths. For sn_single, the default is 1; for sn_all, the default is max.

A typical setting for the High Performance Switch is

```
network.mpi=csss,shared,US
```

#@ node = <*integer*>
Specifies the number of nodes, often used in conjuction with the total_tasks keyword. For example, to request 16 tasks on two nodes, you would specify

```
node = 2
total_tasks = 16
```

The default value for node is 1.

#@ node_usage = shared | not_shared
Specifies whether or not the job may be run on a node with the tasks of other jobs. Generally, the default value of shared should be used, but you may want to request not_shared if your job uses most of the memory of the node.

#@ notification = start | error | complete | always | never
Controls when LoadLeveler will send you mail about your job. You can ask for email notification only when the job starts, only if the job fails, only when it completes (the default), all of the above, or none of the above.

#@ output = <*filename*>
Specifies the name of a file to hold standard output. By default, standard output is directed to /dev/null. See the description of the error keyword for a sample usage.

#@ queue
Places one copy of the job step in the queue. This statement is required. It should be the last keyword statement of the job step.

#@ requirements = <*expression*>
There are a lot of things that can go here; one is to request a specific

node, which might be required if a particular node had been dedicated for your project. Sample syntax is

```
requirements = (Machine == "emerald12")
```

`#@ startdate = <date> <time>`
Specifies a date and time to run the job step. The date is specified as MM/DD/YYYY and the time is specified as hh:mm or hh:mm:ss. By default, the job starts as soon as resources become available.

`#@ total_tasks = <integer>`
Specifies the total number of tasks for a parallel job, used in conjuction with the `blocking` or `node` keywords.

Here's a sample LoadLeveler job command file for a parallel job:

```
#@ error   = err.$(jobid)
#@ output  = out.$(jobid)
#@ notification  = never
#@ environment  = MP_EUILIB=us
#@ wall_clock_limit = 1:00:00
#@ job_type = parallel
#@ node = 2
#@ total_tasks = 16
#@ network.mpi = csss,shared,US
#@ class = staff
#@ queue
linpack
```

The above example requests 16 processors across 2 nodes to run an MPI program called `linpack` in the `staff` queue. The environment variable setting of `MP_EUILIB=us` enables "user-space" adapters for enhanced MPI performance on a node.

The following sample LoadLeveler job script is for a serial job:

```
#@ error   = err.$(jobid)
#@ output  = out.$(jobid)
#@ notification  = error
#@ job_type = serial
#@ class = single_proc
#@ queue
date
```

Note that for a serial job, the keywords `node`, `total_tasks`, and `network.mpi` should not be specified.

4.5.3 Job accounting

Each NESC[2] project has an allocation of CPU time granted by the High Performance Computing Working Group following peer review. If you do work for more than one project, you should make sure that the computing cycles you use are attributed to the correct project by defining the `account_no` keyword in your LoadLeveler job scripts. For example, a job script for work on the STAFF project should contain the line

```
#@ account_no = staff
```

The `myaccount` command will show you which projects may appear on this line for your account. If your account is only active on one project, you may omit the `class` and `account_no` keywords in your job scripts.

To view the fiscal year-to-date usage for your account and each of your projects, enter the NESC^2 command `ru` (short for "resource utilization").

4.5.4 Why isn't my job running?

This is one of the most frequently asked questions at NESC^2. The `llq` and `llclass` commands, particularly the long form of the commands with the `-l` option, may provide the answer. Here are several possibilities:

1. There are not enough processors free to run the job.

   If there aren't enough processors available to run your job, the command

   ```
   llq -s <your_job_Id>
   ```

   will say something like

   > Total number of available initiators of this class on all machines in the cluster: 31
   > Minimum number of initiators of this class required by job step: 32
   > The number of available initiators of this class is not sufficient for this job step.

   The number of available initiators can also be seen in the column labeled "Free Slots" in the output of the `llclass` command.

2. There are not enough initiators left in your class to run the job.

   Each class has its own limit on the number of jobs that can be running and the number of processors that can be in use at one time. Even though there may be resources available on the system, your job may not be able to run until another job in your class finishes. It is usually necessary to look at the output of `llq -l` to see what else is running in your class.

3. You have asked for resources that don't exist.

   LoadLeveler will accept all sorts of invalid requests, such as for queues that don't exist, or for more than 8 processors on a node, or for more than 14 nodes, hoping against hope that some day we will install a machine that can satisfy the request. We should probably filter these out, but for now they are admitted.

4. The log files can not be created.

   If the `llq` command shows your job is in the `(alloc)` state, LoadLeveler may be unable to open the standard error or output files specified in your script. You may have specified an existing file without write permission. You will have to change the permission of the error or output files, or else cancel the job, move the existing log files out of the way, and resubmit.

4.6 Debuggers

The source level debugger available on the NESC^2 HPC systems is called totalview. To use the debugger, you must compile your program with the `-g` option. To run the debugger interactively, enter `totalview` on the command line and choose "Load new program" or "New program" from the File menu. You can also specify the core file or program to run from the `totalview` command line. Online help is available from the TotalView window.

4.7 Data format conversion

Many old datasets can not be read directly on emerald because they were written in binary format on one of the previous NESC^2 systems, either the CRAY C90 (1994-2001) or the CRAY T3E (1999-2004). The IBM systems do not have an equivalent of the `assign` command from the CRAY environment and so do not support automatic data conversion. Instead, any "foreign" data conversions required on input must be done by reading the raw data into a buffer and converting it to the desired format by calling library routines. The NCARU library from

`http://www.scd.ucar.edu/docs/conversion.tools/ncaru.html`

has been installed in `/usr/local/ncaru` for this purpose. For files written in binary format on the CRAY C90, data must be converted from Cray floating point format to IEEE format, which is native on the IBM. For files written in binary format on the CRAY T3E, no data reformatting is necessary because the CRAY T3E used IEEE arithmetic, but the file blocking format still needs to be converted. Please contact the NESC^2 staff if you need assistance with the NCARU library.

5 Archival Storage Management: /asm

Archival storage management at NESC^2 is provided by a Sun E4500 system running SAM-FS, backed by a complex of STK tape silos that provide near-line access to approximately 200 TB of data. To ensure high availability, the archival server consists of two nodes, sweetgum and walnut, operating in a cluster configuration so that either system can fail over to the other. The archival file system is mounted via NFS to emerald. The archival file system is actually two file systems, `/asm1` and `/asm2`, and `/asm/<uid>` (where `<uid>` is a user ID) is a link to either `/asm1/<uid>` or `/asm2/<uid>`. Since all users can reference their files as if they were stored in `/asm/<uid>`, we will refer to `/asm` as one file system for simplicity.

Although `/asm` looks just like a regular UNIX file system and supports UNIX commands to access files stored there, some functions are not advised for performance reasons. For instance, writing to a file on `/asm` is very slow, so the NESC^2 staff recommend creating a file on the locally mounted file systems and copying it to `/asm` when it is complete. In addition, on the IBM eServer, jobs running on the batch nodes must use the local file system for all file I/O because only the interactive nodes (emerald00 and emerald01) have direct access to `/asm`. The NESC^2-specific commands archput and archget have been developed to facilitate file transfers between the local file systems on the NESC^2 HPC systems and `/asm`.

5.1 Putting files to /asm

The NESC[2] command `archput` is the most efficient way to copy local files to `/asm`. The simplest usage is

```
archput <filename>
```

which copies `<filename>` to a corresponding directory on `/asm`. The `archput` command will replicate the current directory structure for `<filename>` on `/asm` by replacing `/home`, `/gpfs/home`, or `/work` in the full path for `<filename>` with `/asm`. For example,

```
/home/gah/<filename>      is copied to  /asm/gah/<filename>
/work/gah/sub/<filename>  is copied to  /asm/gah/sub/<filename>
```

If the file already exists on `/asm`, it is overwritten. You can specify an alternate destination for the file using the `-a` option. The command

```
archput -a newdir <filename>
```

will copy `<filename>` to `/asm/gah/newdir` for user `gah`.

The `archput` command can also take a list of files from standard input for use in a UNIX pipe. For instance, you could copy all the files in the directory `mydir`, including any subdirectories, with the commands

```
find mydir -type f -print | archput
```

Note that the `archput` command copies files to `/asm`, but it doesn't delete the original file. You should verify that the sizes and checksums match before deleting the original file.

5.2 Recalling files from tape

Former sequoia users may recall the Data Migration Facility (DMF) command `dmget` to recall a migrated file from tape. The equivalent command for the `/asm` file system is called `archget`. The `archget` command causes one or more files to be staged from the archive media to online disks for faster access. The syntax is

```
archget <filelist>
```

where `<filelist>` specifies a list of files to recall from `/asm`. Note that this form of the archget command simply recalls files from tape to the disk cache on the archival server; it doesn't copy the files to local disk on the HPC system. As the disk cache fills up, the archival server will release its local copy and the file will have to be recalled from tape again if it is still needed.

5.3 Reading files from /asm

Read performance from `/asm` to the IBM eServer is competitive to that of local disks, so it is best just to leave the file where it is for preprocessing operations that run in the `single_proc` queue on emerald00 and emerald01. However, copying the file to local disk is a necessity when running on the batch nodes because they

do not have direct access to /asm. The archget command with the -c option is used to copy a file from /asm to a local directory:

```
archget -c <cdir> <filelist>
```

If <cdir> is a full path name, the files are copied there; otherwise they are copied to /work/gah/<cdir> for user gah. To copy a whole directory from /asm to local disk, replace <filelist> with the output of the find command:

```
archget -c <cdir> 'find mydir -type f -print'
```

5.4 Old file archives

As systems are retired at NESC^2, their files are archived to /asm. The archival file system currently contains the following repositories of old files:

`/asm/sequoia/home`
: Files that formerly resided in /home on the NESC^2 CRAY C90 system, sequoia (1994--2001)

`/asm/sequoia/work`
: Files that formerly resided in /work on sequoia

`/asm/hickory/home`
: Files that formerly resided in /home on the NESC^2 CRAY T3E system, hickory (1999--2004)

`/asm/hickory/work`
: Files that formerly resided in /work on hickory

`/asm/cypress/home`
: Files that formerly resided in /gpfs/home on the NESC^2 IBM SP system, cypress (2001--2004)

`/asm/cypress/work`
: Files that formerly resided in /work on cypress

`/asm`
: Files that formerly resided in /archive, a file system that was shared between sequoia and hickory from 1999 to 2001. Note that this is also the default location for new files copied to /asm.

5.5 Backup procedures

Only the system files are backed up on the NESC^2 HPC systems. It is not feasible to back up the multi-terabyte user file systems at the present time. All users are encouraged to preserve their most important files by copying them to /asm.

There is only one copy of files sent to tape from /asm. The I-nodes (the file metadata) are backed up several times per day on the archival file server, so if we lose a tape, we can tell you what was on it.

6 Customer Support Services

6.1 Hours of operation

The NESC^2 systems are available 24 hours per day, seven days a week. Operations support is provided 24/7 to ensure that systems problems are identified quickly. The normally scheduled time for system maintenance is Mondays from 5:00 - 8:00 a.m. (Eastern Time). The systems may remain accessible during the maintenance period.

6.2 Who to call for help

The EPA Call Center (866-411-4372) is staffed Monday--Friday, 6:00 am -- 6:30 pm (ET). Call Center staff are not trained on the NESC^2 systems and but may be able to direct your call to a NESC^2 staff member. Inside the EPA domain, the NESC^2 staff may be reached using the e-mail address `help@nesc.epa.gov`. Contact information for the EPA Service Managers can be found in Section 1.3 on page 5.

When the Call Center is not in service, operations problems may be directed to the NESC^2 operators' console: 800-334-2405.

6.3 Finding information on the HPC systems

Several different resources provide online access to system and application documentation for NESC^2 users:

- A message of the day (`/etc/motd`) is displayed when you log on to the NESC^2 systems and may be used to communicate system status information.
- The `news` command provides access to local configuration information and copies of memos sent to the NESC^2 users. Enter `news items` for a list of available topics.
- UNIX `man` pages provide terse online documentation; enter `man man` for more details about how to use the man command.
- NESC^2-specific commands, such as archget and archput, have a help screen accessible with the `-h` option, for example, `archput -h`.
- The module display utility <module help *topic*> may provide details and links to documentation for installed software (see Section 4.2 on page 13).
- The NESC^2 public web page, `http://www.epa.gov/nesc`, contains staff contact information, resource request forms, and documentation such as this guide.
- The NESC^2 internal web page, `http://linden.nesc.epa.gov`, contains additional information for users inside the EPA firewall.
- IBM manuals are available online; one such location is `http://publib.boulder.ibm.com/clresctr/`.

6.4 Code porting and optimization

NESC^2 staff can assist with troubleshooting, code porting, performance evaluation, and optimization. Current account holders may contact Ed Anderson (contractor) at 919.541.0299 or anderson.edward@epa.gov for more information about NESC^2's code porting and optimization services. Non-account holders interested in this service should contact the EPA manager identified in Section 1.3 on page 5.

6.5 Computational chemistry

NESC^2 staff provide computational chemistry expertise. Current account holders may contact Dr. Charles Foley (contractor) at 919.541.1184 or foley.chuck@epa.gov for more information about NESC^2's computational chemistry services. Non-account holders interested in this service should contact the EPA manager identified in Section 1.3 on page 5.

6.6 Computational fluid dynamics

NESC^2 also has on staff specialists in computational fluid dynamics (CFD). These staff are familiar with applying Navier-Stokes equations, available in commercial-off-the-shelf (COTS) applications, such as Fluent, to scientific problems such as fluid through dense urban areas and the human respiratory system. NESC^2 staff can also create custom CFD applications based on available funding. Current account holders may contact Dr. Matt Freeman (contractor) at 919.541.4293 or freeman.matt@epa.gov for more information about NESC^2's CFD services. Non-account holders interested in this service should contact the EPA manager identified in Section 1.3 on page 5.

6.7 Scientific visualization

A key part of the National Environmental Scientific Computing Center is its Scientific Visualization Center (SVC). SVC offers a wide variety of services and these are described on its Web site at http://www.epa.gov/vislab/. Please contact the EPA Service Manager identified in Section 1.3 on page 5 if you are interested in NESC^2's visualization services.

6.8 NESC^2 user training

NESC^2 staff can provide training for users. For information about available training, see the news item *NESC_training*. For any requests concerning training, EPA users should contact the EPA Service Manager.

This User Guide is available in Adobe Portable Document Format (PDF) on US EPA's Web site at

http://www.epa.gov/nesc/10_publications/Customer_Document/UserGuide.pdf

Additional user documentation is also available at

http://www.epa.gov/nesc/10_publications/Customer_Document/

7 List of acronyms

API	Application Programming Interface
ASM	Archival Storage Manager
CFD	Computational Fluid Dynamics
COTS	Commercial off-the-shelf
DAT	Digital Audio Tape
DDS	Digital Data Storage
DMF	Data Migration Facility
EPA	Environmental Protection Agency
GB	Gigabyte (1024 MB)
Gflop/s	Gigaflops (billions of floating-point operations per second)
GPFS	General Parallel File System for AIX
HPC	High Performance Computing
HPCWG	High Performance Computing Working Group
IBM	International Business Machines
IEEE	Institute of Electrical and Electronics Engineers
KB	Kilobyte (1024 bytes)
MB	Megabyte (1024 KB)
Mflop/s	Megaflops (millions of floating-point operations per second)
MHz	Megahertz (millions of cycles per second)
$NESC^2$	National Environmental Scientific Computing Center
NOAA	National Ocean and Atmospheric Administration
PC	Personal Computer
POE	Parallel Operating Environment
RTP	Research Triangle Park
SVC	Scientific Visualization Center
TB	Terabyte (1024 GB)
TSSMS	Time Sharing Services Management System